

ONE PAGE SUMMARY

IT & Software > Software Architecture > Hexagonal Architecture

Introduction to Hexagonal Architecture

Hexagonal Architecture: Structuring Software for Flexibility and Testability

DEFINITION

Hexagonal Architecture, also known as Ports and Adapters, is a software design pattern that separates an application's core logic from external systems. This separation allows for easier testing and adaptation to changes in technology or user requirements.

KEY CONCEPTS

Core Domain

The central part of the application that contains the business logic and rules.

Ports

Interfaces that define how the core domain interacts with external systems.

Adapters

Implementations that connect the core domain to external systems, such as databases or APIs.

Isolation

The architecture isolates the application logic from external changes, enhancing maintainability.

Testability

The design allows for easy unit and integration testing of the core logic without external dependencies.

EXAMPLES

- A web application that uses REST APIs as ports to communicate with the core logic.
- A payment processing system where the core logic remains unchanged regardless of the payment gateway used.
- An e-commerce platform that can switch between different databases without affecting business rules.

MEMORY TIPS

- ★ Think of 'hexagon' as a shape with multiple sides, representing different external systems.
- ★ Remember 'Ports and Adapters' to visualize how core logic connects to the outside world.
- ★ Use 'Isolation for Flexibility' to recall the main benefit of this architecture.

COMMON MISTAKES

- ✗ Confusing adapters with ports; remember, ports define interfaces, while adapters implement them.
- ✗ Neglecting to properly isolate the core domain, leading to tight coupling with external systems.
- ✗ Overcomplicating the architecture by adding unnecessary layers instead of keeping it simple.

QUICK RECAP

Hexagonal Architecture separates core logic from external systems using ports and adapters. This design enhances flexibility and testability, making it easier to adapt to changes and maintain the application.